Paleti Krishnasai
CED18I039

Operating Systems - Assignment 5 - PIPES

1. Test Drive codes discussed in class.

Code 1:

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

#define BUFFER_SIZE 25
#define READ_END 0
#define WRITE_END 1

int main(void)
{
   char write_msg[BUFFER_SIZE];
   scanf("%[^\n]%*c", write_msg);
   char read_msg[BUFFER_SIZE];
   int fd[2];
   pid_t pid;
 // pipe creation
   if(pipe(fd)==-1)
   {
       fprintf(stderr,"Pipe Failed");
       return 1;
   }
   pid = fork();
   if(pid<0)
   {
       fprintf(stderr,"Fork Failed");
       return 1;
   }
```

```
    if (pid>0)
    {
        // parent process
        // close the unused end of the pipe
        close(fd[READ_END]);

        //write to the pipe
        write(fd[WRITE_END],write_msg,strlen(write_msg)+1);

        //close the write end of the pipe
        close(fd[WRITE_END]);
    }
    else
    {
        //child process
        //close the unused end of the pipe
        close(fd[WRITE_END]);

        //read from the pipe
        read(fd[READ_END],read_msg,BUFFER_SIZE);
        printf("read %s\n",read_msg);

        //close the read end of the pipe
        close(fd[READ_END]);
    }
    return 0;


}
```

```
paleti@paletil:~/OS_LAB/Practice_Testing$ ./pipes
test message
read test message
paleti@paletil:~/OS_LAB/Practice_Testing$
```

Paleti Krishnasai
CED18I039

Code 2:

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int main()
{
   int pipefd1[2],pipefd2[2];
   int returnstatus1,returnstatus2;
   int pid;
   char pipe1message[20]="pipe1 message";
   char pipe2message[20]="pipe2 message";
   char readmessage[20];
   returnstatus1 = pipe(pipefd1);
   if(returnstatus1 == -1)
   {
       printf("Unable to create pipe 1 \n");
       return 1;
   }
   returnstatus2 = pipe(pipefd2);
   if(returnstatus2 == -1)
   {
       printf("Unable to create pipe2 \n");
       return 1;
   }
   pid = fork();
   if(pid>0)
   {
       close(pipefd1[0]); // close read end of pipe 1
       close(pipefd2[1]); // close write end of pipe 2
       printf("Parent : Writing to pipe 1 - message is
%s\n",pipe1message);
       write(pipefd1[1],pipe1message,sizeof(pipe1message)+1);
       read(pipefd2[0],readmessage,sizeof(readmessage));
       printf("Parent : Reading form pipe 2 - Message is
%s\n",readmessage);
```

Paleti Krishnasai
CED18I039

```
    }

    else
    {
        close(pipefd1[1]); //close write end of pipe 1
        close(pipefd2[0]); //close read end of pipe 2
        read(pipefd1[0],readmessage,sizeof(readmessage));
        printf("Child : Reading from pipe 1 - message is
%s\n",readmessage);
        printf("Child : Writing to pipe 2 - message is %s\n",pipe2message);
        write(pipefd2[1],pipe2message,sizeof(pipe2message)+1);
    }

    return 0;

}
```

```
paleti@paletil:~/OS_LAB/Practice_Testing$ ./pipes2
Parent : Writing to pipe 1 - message is pipe1 message
Child : Reading from pipe 1 - message is pipe1 message
Child : Writing to pipe 2 - message is pipe2 message
Parent : Reading form pipe 2 - Message is pipe2 message
```

Paleti Krishnasai
CED18I039

2. **Parent sets up a string which is read by child, reversed there and read back to the parent .**
3. **String reversal and palindrome check using pipes / shared memory.**

Code (combined both in a single code) :

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

void reverseStr(char* str)
{
   for (int i = 0; i < strlen(str) / 2; i++)
   {
       char temp = str[i];
       str[i] = str[strlen(str) - i - 1];
       str[strlen(str) - i - 1] = temp;
   }
}


int main()

{
   int pipe1[2],pipe2[2];
   int pid;
   int length;
   char inputstr[512];
   printf("Enter the string :\n");
   scanf("%[^\n]%*c",inputstr);
   char readmessage[512];
   char writemessage[512];
   int returnstatus1, returnstatus2;
   returnstatus1 = pipe(pipe1);
   if (returnstatus1 == -1)
```

```
    {
        printf("Unable to create pipe 1 \n");
        return 1;
    }
    returnstatus2 = pipe(pipe2);
    if (returnstatus2 == -1)
    {
        printf("Unable to create pipe2 \n");
        return 1;
    }

    pid = fork();
    if(pid > 0)
    {
        close(pipe1[0]); //close read of pipe1
        close(pipe2[1]); //close write of pipe2
        write(pipe1[1],inputstr,sizeof(inputstr)+1);
        read(pipe2[0],readmessage,sizeof(readmessage));
        printf("Reversed String is  :%s \n",readmessage);
        length=strlen(readmessage);
        if(strncmp(inputstr,readmessage,length)==0)
        printf("%s is a Palindrome\n",inputstr);
        else printf("%s is NOT a Palindrome\n",inputstr);

    }
    else
    {
        close(pipe1[1]); // close write of pipe1
        close(pipe2[0]); //close read of pipe2
        read(pipe1[0],writemessage,sizeof(writemessage));
        reverseStr(writemessage);
        printf("%s\n",writemessage);
        write(pipe2[1], writemessage, sizeof(writemessage) + 1);
    }

    return 0;
}
```

Paleti Krishnasai
CED18I039

Output : reversal and palindrome check

```
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./strrev
Enter the string :
paleti
itelap
Reversed String is  :itelap
paleti is NOT a Palindrome
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./strrev
Enter the string :
lool
lool
Reversed String is  :lool
lool is a Palindrome
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./strrev
Enter the string :
hello kids
sdik olleh
Reversed String is  :sdik olleh
hello kids is NOT a Palindrome
paleti@paletil:~/OS_LAB/Pipes_Threads$
```

Paleti Krishnasai
CED18I039

4. **Parent sets up string 1 and child sets up string 2. String 2 concatenated to string 1 at parent end and then read back at the child end.**

Code :

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int main()

{
   int pipe1[2], pipe2[2];
   int pid;
   char string1[512], string2[512] ,destination[512];
   printf("Enter string1 :\n");
   scanf("%[^\n]%*c", string1);
   printf("Enter string2 :\n");
   scanf("%[^\n]%*c", string2);
   strcpy(destination,string1);
   char readmessage[512];
   char readmessage2[512];
   int returnstatus1, returnstatus2;
   returnstatus1 = pipe(pipe1);
   if (returnstatus1 == -1)
   {
       printf("Unable to create pipe 1 \n");
       return 1;
   }
   returnstatus2 = pipe(pipe2);
   if (returnstatus2 == -1)
   {
       printf("Unable to create pipe2 \n");
       return 1;
   }
```

Paleti Krishnasai
CED18I039

```
    pid = fork();
    if (pid>0)
    {
        close(pipe1[1]); //close write of pipe1
        close(pipe2[0]); //close read of pipe2

        read(pipe1[0], readmessage, sizeof(readmessage));
        strcat(destination,readmessage);
        write(pipe2[1],destination,sizeof(destination)+1);

    }
    else
    {
        close(pipe1[0]); //close read of pipe1
        close(pipe2[1]); //close write of pipe2

        write(pipe1[1],string2,sizeof(string2)+1);
        read(pipe2[0],readmessage2,sizeof(readmessage2));
        printf("Concatnated string is : %s\n",readmessage2);
    }


}
```

Paleti Krishnasai
CED18I039

Output:

```
paleti@paletil:~/OS_LAB/Pipes_Threads$ make strconcat
cc     strconcat.c   -o strconcat
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./strconcat
Enter string1 :
hello
Enter string2 :
   assignment
Concatnated string is : hello   assignment
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./strconcat
Enter string1 :
how are you?
Enter string2 :
 im good
Concatnated string is : how are you? im good
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./strconcat
Enter string1 :
1234
Enter string2 :
567789
Concatnated string is : 1234567789
paleti@paletil:~/OS_LAB/Pipes_Threads$
```

Paleti Krishnasai
CED18I039

5. **Substring generation at child end of a string setup at parent process end.**

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int main()
{
   srand(time(0));
   int pipe1[2], pipe2[2],pipe3[2];
   int returnstatus1, returnstatus2, returnstatus3;
   int pid;
   char inputstring[512],readmessage[512];
   int startindex,endindex,length,start,end;
   printf("Enter the string : ");
   scanf("%[^\n]%*c", inputstring);
   length = strlen(inputstring);



   returnstatus1 = pipe(pipe1);
   if (returnstatus1 == -1)
   {
       printf("Unable to create pipe 1 \n");
       return 1;
   }
   returnstatus2 = pipe(pipe2);
   if (returnstatus2 == -1)
   {
       printf("Unable to create pipe2 \n");
       return 1;
   }
```

```
    returnstatus3 = pipe(pipe3);
    if (returnstatus3 == -1)
    {
        printf("Unable to create pipe 3 \n");
        return 1;
    }
    pid = fork();

    if(pid>0)
    {
        close(pipe1[0]); // close read of pipe 1
        close(pipe2[0]); // close read of pipe 2
        close(pipe3[0]); // close read of pipe 3
        startindex = rand() % length/2;
        endindex = length/2 + rand() % length/2;
        printf("\n%d %d\n",startindex,endindex);

        write(pipe1[1],inputstring,sizeof(inputstring));
        write(pipe2[1],&startindex,sizeof(startindex));
        write(pipe3[1],&endindex,sizeof(endindex));

    }
    else
    {
        close(pipe1[1]); // close write of pipe 1
        close(pipe2[1]); // close write of pipe 2
        close(pipe3[1]); // close write of pipe 3

        read(pipe1[0],readmessage,sizeof(readmessage));
        read(pipe2[0],&start,sizeof(start));
        read(pipe3[0],&end,sizeof(end));
        printf("Substring : ");
        for(int i =start;i<=end;i++)
        printf("%c",inputstring[i]);
        printf("\n");
    }

    return 0;
}
```

Paleti Krishnasai
CED18I039

```
paleti@paletil:~/OS_LAB$ cd Pipes_Threads/
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./substring
Enter the string : operating systems

0 8
Substring : operating
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./substring
Enter the string : operating systems

5 16
Substring : ting systems
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./substring
Enter the string : online is bad

4 7
Substring : ne i
paleti@paletil:~/OS_LAB/Pipes_Threads$ ./substring
Enter the string : online is bad

6 7
Substring :  i
paleti@paletil:~/OS_LAB/Pipes_Threads$
```